

displaying a node in a graphical program, wherein the node is configurable to perform a plurality of operations depending upon user input specifying configuration information for the node;

displaying a graphical user interface (GUI) for specifying configuration information for the node, wherein the GUI comprises information useable in configuring the node to perform one or more operations from the plurality of operations;

receiving user input via the GUI specifying one or more desired operations for the node from the plurality of operations; and

programmatically generating graphical source code for the node to implement the one or more desired operations, in response to the user input.

15. (Amended) A method for configuring a node in a graphical program, the method comprising:

displaying a node in a graphical program, wherein the node is configurable to perform functionality depending upon user input specifying configuration information for the node;

displaying a graphical user interface (GUI) for specifying configuration information for the node, wherein the GUI is useable to specify functionality for the node;

receiving user input via the GUI specifying desired functionality for the node; and programmatically generating graphical source code for the node to implement the specified functionality, in response to the user input.

17. (Amended) A memory medium comprising program instructions for configuring a node in a graphical program, wherein the program instructions are executable to:

display a node in a graphical program, wherein the node is configurable to perform a plurality of operations depending upon user input specifying configuration information for the node;

display a graphical user interface (GUI) for specifying configuration information for the node, wherein the GUI comprises information useable in guiding a user in configuring the node to perform one or more operations from the plurality of operations;

receive user input via the GUI specifying one or more desired operations for the node from the plurality of operations; and

programmatically generate graphical source code for the node to implement the one or more desired operations, in response to the user input.

26. (Amended) A memory medium comprising program instructions for configuring a node in a graphical program, wherein the program instructions are executable to:

display a node in a graphical program, wherein the node is configurable to perform functionality depending upon user input specifying configuration information for the node;

display a graphical user interface (GUI) for specifying configuration information for the node, wherein the GUI is useable to specify functionality for the node;

receive user input via the GUI specifying desired functionality for the node; and

programmatically generate graphical source code for the node to implement the specified functionality, in response to the user input.

Please add the following new claims:

—28. (New) The method of claim 1, further comprising:

displaying the programmatically generated graphical source code in the graphical program.

29. (New) The method of claim 1,

wherein said programmatically generating the graphical source code for the node comprises displaying the programmatically generated graphical source code in place of the node in the graphical program.

30. (New) The method of claim 1, further comprising:
receiving user input selecting the node prior to said displaying the node in the
graphical program

31. (New) The method of claim 1,
wherein the programmatically generated graphical source code comprises a
plurality of interconnected nodes that visually indicate functionality of the graphical
source code.

32. (New) The method of claim 1,
wherein the programmatically generated graphical source code comprises a
plurality of nodes interconnected in one or more of a data flow, control flow, and/or
execution flow format.

33. (New) The method of claim 1,
wherein the programmatically generated graphical source code comprises a
plurality of nodes interconnected to indicate data flow among the nodes.

34. (New) The method of claim 1,
wherein the programmatically generated graphical source code comprises a
plurality of nodes interconnected to indicate control flow among the nodes.

35. (New) The method of claim 1,
wherein the programmatically generated graphical source code comprises a
plurality of nodes interconnected to indicate execution flow among the nodes.

36. (New) The memory medium of claim 26, wherein the program
instructions are further executable to display the programmatically generated graphical
source code in the graphical program.

37. (New) The memory medium of claim 26,
wherein, in programmatically generating the graphical source code for the node,
the program instructions are executable to programmatically generate the graphical
source code as a sub-program of the graphical program, wherein the node represents the
sub-program.

38. (New) The memory medium of claim 26,
wherein, in programmatically generating the graphical source code for the node,
the program instructions are executable to display the programmatically generated
graphical source code in place of the node in the graphical program.

39. (New) The memory medium of claim 26, wherein the program
instructions are further executable to receive user input selecting the node prior to said
displaying the node in the graphical program.

40. (New) The memory medium of claim 26,
wherein the programmatically generated graphical source code comprises a
plurality of interconnected nodes that visually indicate functionality of the graphical
source code.

41. (New) The memory medium of claim 26,
wherein the programmatically generated graphical source code comprises a
plurality of nodes interconnected in one or more of a data flow, control flow, and/or
execution flow format.

42. (New) The memory medium of claim 26,
wherein the programmatically generated graphical source code comprises a
plurality of nodes interconnected to indicate data flow among the nodes.

43. (New) The memory medium of claim 26,
wherein the specified functionality is first functionality;

wherein the program instructions are further executable to:

receive user input requesting to change configuration information for the node, after said programmatically generating the graphical source code for the node;

re-display the graphical user interface (GUI) in response to the user input requesting to change configuration information for the node;

receive user input via the GUI specifying second functionality for the node; and

programmatically generate new graphical source code for the node in place of the previously generated graphical source code, wherein the new graphical source code implements the second functionality.

44. (New) The memory medium of claim 43,

wherein the program instructions are further executable to display the new graphical source code in place of the previously generated graphical source code.

45. (New) The memory medium of claim 26,

wherein no functionality is set for the node until after the graphical source code is programmatically generated for the node.

46. (New) The memory medium of claim 26,

wherein default functionality is set for the node;

wherein said programmatically generating graphical source code for the node comprises replacing the default functionality with functionality implemented by the programmatically generated graphical source code.

47. (New) The memory medium of claim 26,

wherein no program instructions to be executed during execution of the graphical program are associated with the node until after graphical source code is programmatically generated for the node.

48. (New) A method for creating a graphical program, the method comprising:

selecting a graphical program node in response to user input;

displaying a graphical user interface (GUI) for configuring operation of the graphical program node;

receiving user input to the GUI configuring desired operation of the graphical program node;

programmatically generating graphical source code based on the user input configuring desired operation of the graphical program node; and

displaying the programmatically generated graphical source code.

49. (New) The method of claim 48,

wherein said displaying the programmatically generated graphical source code comprises displaying the programmatically generated graphical source code in place of the node in the graphical program.

50. (New) The method of claim 48,

wherein said receiving user input to the GUI configuring desired operation of the graphical program node comprises receiving user input to the GUI configuring first operation of the graphical program node;

wherein the method further comprises:

receiving user input requesting to change operation of the node, after said programmatically generating the graphical source code;

re-displaying the graphical user interface (GUI) in response to the user input requesting to change operation of the node;

receiving user input to the GUI configuring second operation of the graphical program node; and

programmatically replacing the previously generated graphical source code with new graphical source code, wherein the new graphical source code implements the second operation.

51. (New) The method of claim 48, further comprising:
displaying the graphical program node in the graphical program in response to
said selecting.

52. (New) The method of claim 48,
wherein the programmatically generated graphical source code comprises a
plurality of interconnected nodes that visually indicate functionality of the graphical
source code.

53. (New) The method of claim 48,
wherein the programmatically generated graphical source code comprises a
plurality of nodes interconnected in one or more of a data flow, control flow, and/or
execution flow format.

54. (New) The method of claim 48,
wherein the programmatically generated graphical source code comprises a
plurality of nodes interconnected to indicate data flow among the nodes.

55. (New) The method of claim 48,
wherein said programmatically generating the graphical source code comprises
programmatically generating the graphical source code as a sub-program of the node.

56. (New) The method of claim 48,
wherein the GUI comprises at least one panel.

57. (New) The method of claim 48,
wherein the GUI comprises a plurality of panels.

58. (New) A method for creating a graphical program, the method
comprising:

wherein said displaying the programmatically generated graphical source code comprises displaying the programmatically generated graphical source code in place of the node in the diagram.

62. (New) The memory medium of claim 60,
wherein said displaying the programmatically generated graphical source code comprises displaying the programmatically generated graphical source code in the diagram;

wherein the graphical program node is no longer displayed in the diagram at least as of when the programmatically generated graphical source code is displayed in the diagram.

63. (New) The memory medium of claim 60,
wherein said displaying the programmatically generated graphical source code comprises displaying the programmatically generated graphical source code in the diagram;
wherein the program instructions are further executable to implement:
discontinuing displaying the graphical program node in the diagram after displaying the programmatically generated graphical source code in the diagram.

64. (New) The memory medium of claim 60,
wherein the programmatically generated graphical source code comprises a plurality of interconnected nodes that visually indicate functionality of the graphical source code.

65. (New) The memory medium of claim 60,
wherein the programmatically generated graphical source code comprises a plurality of nodes interconnected in one or more of a data flow, control flow, and/or execution flow format.--